FILED ELECTRONICALLY IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Title	:	Sorting Objects Having Variable Length Keys			
Inventor	:	Christopher Y. Blaicher			
Serial No	:	10/618,500	Docket	:	149-0105US (02-014-US)
Examiner	:	Usmaan Saeed	Art Unit	:	2166
Customer	:	29855			

APPEAL BRIEF

This is an appeal from the final rejection of claims 1-63 as set forth in the Examiner's final Office Action (dated 15 May 2006).

Table of Contents

Table of Authorities

Cases

Statutes

Other Authorities

REAL PARTY IN INTEREST

The real party in interest in the above-referenced application is BMC Software, Inc. of Houston, Texas.

RELATED APPEALS AND INTERFERENCES

To the present knowledge of Appellants’ representative, there are currently no related appeal or interference proceedings that will directly affect, or be directly affected by, or have a bearing on, the Board’s decision in the present Appeal.

STATUS OF CLAIMS

Claims 1-63 stand rejected. Claims 64-69 have been withdrawn (pursuant to a restriction requirement under 35 U.S.C. 121). Claims 1-63 are appealed.

STATUS OF AMENDMENTS

No amendments have been filed subsequent to the Examiner’s final Office Action dated 15 May 2006.

SUMMARY OF CLAIMED SUBJECT MATTER

Independent claims 1 and 16 are directed to methods for sorting an object having variable length key information (page 1 at lines 4-6 in ¶ 0001 and page 5 at lines 11-12 in ¶ 0015) comprising the acts of:

- “obtaining a plurality of data records” as described at page 3 (line 20 in ¶ 0006), page 5 (lines 21-22) and FIG. 3 at block 305.
- “for each data record

  - extracting key information" as described at page 6 (lines 2-6 in ¶ 0016), FIG. 3 at block 310, page 7 (lines 8-16 in ¶ 0018) and FIG. 4 at blocks 415-430. One method to extract variable length key information using an illustrative sort control card is described at page 9 (line 4 in ¶ 0021) to page 11 (line 9 in ¶ 0024) and FIG. 5. Another illustrative technique to extract, pad and sort variable length keys is described at page 13 (lines 11-17 in ¶ 0028).
  - "and storing the expanded key information in a key record" as described at page 6 (lines 2-6 in ¶ 0016), page 7 (lines 16-18 in ¶ 0018) and FIG. 4 at block 435.
  - "wherein the expanded key information is not stored in intermediate storage" as described at page 6 (lines 19-21 in ¶ 0017) and page 8 (lines 20-22 in ¶ 0020).

- "sorting the plurality of key records based on the expanded key information" as described at page 6 (lines 9-12 in ¶ 0016), FIG. 3 at block 315 and page 8 (lines 12-14 in ¶ 0020).
- "reorganizing the plurality of data records to correspond to the order of the sorted plurality of key records" as described at page 6 (lines 12-14 in ¶ 0016), FIG. 3 at block 320 and page 8 (lines 14-16 in ¶ 0020).
- "and storing the reorganized plurality of data records without their associated expanded key information to a working storage" as described at page 7 (lines 1-7 in ¶ 0017), FIG. 3 at block 335, page 11 (line 17 in ¶ 0026) to page 12 (line 6 in ¶ 0026) and FIG. 6.

A specific embodiment of the claimed invention as it may be used to reorganize a DB2® database is described at page 12 (line 18) to page 13 (line 10 in ¶ 0027) and FIG. 7. (DB2 is a registered trademark of International Business Machines corporation.)

Independent claim 27 is directed to a program storage device having instructions stored thereon for performing the acts of independent claim 1 as described above and at page 4 (lines 7-8 in ¶ 0006), page 14 (line 21) to page 15 (line 1 in ¶ 0030) and page 15 (lines 6-11 in ¶ 0030).

Independent claim 40 is directed to a sorting system having memory means (as described at page 7 at lines 8-11 in ¶ 0018; FIG. 4 at element 400; page 7 at lines 13-15 in ¶ 0018; page 8 at lines 16-20 in ¶ 0020; page 11 at lines 17-19 in ¶ 0026; FIG. 6 at elements 600, 605, 610 and 630; and page 14 at lines 8-11 in ¶ 0029) and processing means (as described at page 14 at line 21 to page 15, line 6 in ¶ 0030) for performing the acts of independent claim 1 as described above.

Independent claim 55 is directed to a method to sort an object having variable length keys in accordance with independent claim 1 (see citations above) with the added limitations of "repeating the acts of obtaining, sorting, reorganizing and storing for at least a second plurality of data records" and "merging the at least two plurality of reorganized data records" as described at page 5 (line 10 in ¶ 0016) to page 7 (line 7 in ¶ 0017) in conjunction with FIG. 3 at blocks 305-330, page 12 (lines 6-12 in ¶ 0026), FIG. 6, page 13 (lines 6-8 in ¶ 0027) and FIG. 7 at block 730.

GROUNDS FOR REJECTION TO BE REVIEWED ON APPEAL

Whether claims 1-12, 16-24, 27-37 and 40-52 are unpatentable under 35 U.S.C. 103(a) over Applicant's Admitted Prior Art ("AAPA") in view of U.S. Patent 5,247,665 to Matsuda et al. ("Matsuda").

Whether claims 13-15, 25, 26, 38, 39 and 53-63 are unpatentable under 35 U.S.C. 103(a) over AAPA in view of Matsuda and further in view of U.S. Patent 5,274,805 to Ferguson et al. ("Ferguson").

ARGUMENTS

1. Subject Matter Disclosed by AAPA

AAPA describes a prior art technique to sort objects having variable length key information. Specification at page 1, line 21 to page 2, line 10 (¶ 0003) and FIG. 1. As described there, an object's records including variable length key information are obtained, the key values are padded or expanded to a fixed length and the records (including the expanded key values) are stored to intermediate storage. Once all of the object's records have been processed in this manner, the records (including the padded/expanded key information) are retrieved from intermediate storage and sorted. Sorted records have their key values unpadded or compressed and then loaded back into the object.

Identified drawbacks to the described prior art approach include the fact that because "many keys are a fraction of the size that must be supported; the padded copy of an object can be several times the size of the original. In addition, since a single object can not generally be retained in working memory, the time required to write and read an intermediate file having expanded sort keys can consume a significant portion of the total time needed to sort the object." Specification at page 2, lines 10-18 (¶ 0003). One known prior art method to mitigate these drawbacks includes the use of E15 and E35 programs. Specification at page 2, line 19 to page 3, line 8 (¶ 0004) and FIG. 2. As explicitly described, however, such approaches continue to require that records including padded/expanded key information be written to intermediate storage. In addition, the described techniques do not separate expanded key information from the target data records. Specification at Page 2, line 19 to page 3, line 8 (¶ 0004) and FIG. 2.

2. Subject Matter Disclosed by Matsuda

Matsuda describes ten (10) embodiments of a database processing apparatus to, *inter alia*, perform sorting operations in a relational database. Matsuda at Abstract, 1:62-2:2, 2:3-33 (first embodiment), 2:34-3:2 (second embodiment), 3:3-22 (third embodiment), 3:23-56 (fourth embodiment), 3:57-4:24 (fifth embodiment), 4:25-59 (sixth embodiment), 4:60-5:25 (seventh embodiment), 5:26-64 (eighth embodiment), 5:65-6:35 (ninth embodiment) and 6:36-7:17 (tenth embodiment).[1] The disclosed key manipulation operation is consistent across all embodiments. Specifically with respect to the embodiment relied upon by the Examiner (embodiment 10), Matsuda describes "extracting a key … from each record of the … target file … [that has been] … stored in … main memory" (Matsuda at 6:52-54), "adding a serial number or relative position data of a record having the key to the key" (Matsuda at 6:55-57), using the (modified) key value to obtain data, performing a operation on the obtained data based on a specified command and storing the data (Matsuda at 6:58-64).

3. Subject Matter Disclosed by Ferguson

Ferguson is directed to "sorting and compressing data that has particular advantages in implementing a key index tree structure." Ferguson at 1:15-17. More specifically, sort methods in accordance with Ferguson use "substrings to sort strings of key records into a linked list structure that can be directly transformed into an index tree." Ferguson 4:39-41. In Ferguson, sorting "is carried out in two phases, consisting of a pre-sort phase (similar to the prior art described by Appellant), followed by one or more merge passes that take advantage of the concept of a 'substring'." Ferguson at 7:4-7. "In the pre-sort phase, a file of data records on a storage system is read and key records are extracted in known fashion … sorted [and] … written back to the storage

[1] As used herein, the notation C:A-B means column C, Lines A through B. The notation C:A-D:B means column C, line A to column D, line B.

system … [where] … as the sorted key records are written out onto the storage system, a 'substring field' is inserted at intervals in the data string to delimit the output string into substrings." Ferguson at 7:8-30. *See also* Ferguson at 14:50-15:2 and FIG. 9 (elements 90-94). During the merge phase, "[s]ubstrings are read from the storage system into associated input buffers reserved in the computer memory." Ferguson at 8:34-36. "Merge comparisons are performed on the contents of input buffer[s] … with the sorted output records being stored in the output buffer. When the output buffer becomes full, it is written out to" storage. Ferguson at 8:67-9:4 and FIGS. 3-5.

4. Legal Principles

To establish a *prima facie* case of obviousness, three criterion must be met: (1) the cited prior art references must teach or suggest all of the claimed limitations; (2) there must be some suggestion or motivation to make the combination; and (3) there must be a reasonable expectation of success. *In re Vaeck*, 947 F.2d 488, 493, 20 U.S.P.Q.2d (BNA) 1438, 1442 (Fed. Cir. 1991); *see also* M.P.E.P. 2143. With respect to the required *motivation to combine* element, "[b]oth the suggestion and the reasonable expectation of success must be founded in the prior art, not in the applicant's disclosure." *In re Vaeck*, 947 F.2d 488, 493, 20 U.S.P.Q.2d (BNA) 1438, 1442 (Fed. Cir. 1991); *see also* M.P.E.P. 2143.

5. Independent Claims 1, 16, 27 and 40 are not Rendered Obvious by the Combination of AAPA and Matsuda

A. Summary of Claimed Subject Matter

Independent claims 1, 16, 27 and 40 are directed to methods (claims 1 and 16), a program storage device (claim 27) and a system (claim 40) for sorting an object's records by extracting, expanding and storing variable length key information from each of the object's records into key records where the *key records are not stored in intermediate storage*. The key records are then sorted, the object's data records are reorganized in accordance with the sorted key records and the reorganized/sorted data records are thereafter stored (without their expanded key information).

B. Examiner's Rejection

The Examiner relies on AAPA as teaching all of the claimed elements except for "wherein the expanded key information is not stored in intermediate storage," relying on Matsuda for this limitation. Office Action dated 15 May 2006 at page 5, second and fourth paragraphs. In addition, and in response to Appellant's earlier communications, the Examiner asserts that only independent claim 16 recites sorting data objects having variable length keys. Advisory Action dated 3 August 2006 at Continuation Page, lines 3-6.

C. Analysis

As an initial matter, the Examiner's contention that independent claims 1, 27 and 40 are not directed to sorting data objects having variable length keys is without merit. Independent method claim 1 recites "expanding [] extracted key information." Similarly, independent claims 27 and 40 recite instructions for causing a programmable control device (claim 27) and processing means (claim 40) to "expand [] extracted key information." First, the act of "expanding" does not make sense in light of the Specification and invention if the key information is not variable length. As such, independent claims 1, 27 and 40 inherently recite data records whose key information is variable length. Second, as taught by the specification and acknowledged by the Examiner, "[t]he act of padding converts variable length key fields to fixed length key fields of a size great enough to accommodate any value that the key may assume." Specification at page 2, lines 1-3 (¶ 0003) and Advisory Action dated 3 August 2006 at Continuation Page, lines 5-6. Finally, the Specification uses the term "expand" synonymously with the term "pad." See, for example, Specification at page 2, lines 3-4 ("Once padded, the record is written to an intermediate file …") and page 2, lines 15-17 ("… the time required to write and read an intermediate file having expanded sort keys

…") where the two terms are used in the context of a single prior art example. See also page 12, line 20 to page 13, line 2 and FIG. 7 ("Data is retrieved … [and] … Each record is passed to E15 interface 705 which … extracts and expands … key fields, storing the padded keys as fixed length components in a key record stored in key structure 715"). Appellant's clear teaching that "padding" converts variable length key information to fixed length key information along with its consistent and synonymous use of the terms pad and expand make it undeniably clear that the each of claims 1, 27 and 40 are directed to operations involving data records having variable length key field(s).

With respect to Matsuda, nowhere does Matsuda teach or even suggest the use of variable length key information. Key manipulation operations as taught by Matsuda, and relied upon by the Examiner, involve adding an offset (a memory address) to an initial key value to obtain an address from which data is obtained. See, for example, Matsuda at 8:40-49, 8:64-9:3, 9:56-62, 12:26-34 and 13:15-24. Thus, it is not possible for Matsuda to use variable length keys. If this were not so, there would be no way to uniformly generate an address – the length of which must be a fixed size. Matsuda confirms this interpretation. Matsuda at 9:3-8 ("… since each record comprises a plurality of key fields, and each record length and each key field length are constant, the start location data (on the main memory 9) of each record can be obtained by multiplying the record length by the number of records."). Thus, *not only does Matsuda not contemplate the use of variable length key information, the techniques taught therein do not suggest or even accommodate use of variable length key information*.

Accordingly, even if AAPA and Matsuda could be interpreted as the Examiner contends, these references do not suggest their combination – especially in light of the recognition that Matsuda does not seem to permit the use of data records having variable length keys. On this point the Examiner argues that "[i]t would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the claimed references because Matsuda's teaching would have allowed AAPA to reduce the load factors of the CPU and the bus system and the processing performance is greatly improved by not transferring the result from main memory to the magnetic disk/intermediate storage." Office Action dated 15 May 2006 at page 5, paragraphs 3 and 4.

In essence, the Examiner is arguing that because it would solve a known problem to combine the references, it is obvious to do so.

Such reasoning is not of the type or quality required to make a *prima facie* case of obviousness. *In re Gordon*, 733 F.2d 900, 902, 221 U.S.P.Q. (BNA) 1125, 1127 (Fed. Cir. 1984) ("The mere fact that the prior art could be so modified would not have made the modification obvious unless the prior art suggested the desirability of the modification."); *In re Dembiczak*, 175 F.3d 994, 999, 50 U.S.P.Q.2d (BNA) 1614, 1617 (Fed. Cir. 1999) ("Our case law makes clear that the best defense against the subtle but powerful attraction of a hindsight-based obviousness analysis is rigorous application of the requirement for a showing of the teaching or motivation to combine prior art references. See, *e.g.*, *C.R. Bard, Inc. v. M3 Sys., Inc.*, 157 F.3d 1340, 1352, 48 U.S.P.Q.2D (BNA) 1225, 1232 (Fed. Cir. 1998) (describing "teaching or suggestion or motivation [to combine]" as an "essential evidentiary component of an obviousness holding"); *In re Rouffet*, 149 F.3d 1350, 1359, 47 U.S.P.Q.2D (BNA) 1453, 1459 (Fed. Cir. 1998) ("the Board must identify specifically … the reasons one of ordinary skill in the art would have been motivated to select the references and combine them"); *In re Fritch*, 972 F.2d 1260, 1265, 23 U.S.P.Q.2D (BNA) 1780, 1783 (Fed. Cir. 1992) (examiner can satisfy burden of obviousness in light of combination "only by showing some objective teaching [leading to the combination]"); *In re Fine*, 837 F.2d 1071, 1075, 5 U.S.P.Q.2D (BNA) 1596, 1600 (Fed. Cir. 1988) (evidence of teaching or suggestion "essential" to avoid hindsight).

For at least this reason, the Examiner has failed to state a *prima facie* case of obviousness under 35 U.S.C. 103(a), the elements of which are identified in Section 4 above.

6. Independent Claim 55 is not Rendered Obvious by the Combination of AAPA, Matsuda and Fergusson

A. Summary of Claimed Subject Matter

Independent claim 55 is a method directed to substantially the same subject matter as are independent claims 1, 16, 27 and 40 (see Section 5.C above) except that it further requires a repeated application of the extract, expand, sort, reorganize and store operations described above with respect to independent claims 1, 16, 27 and 40, for a second group of data records. See "Summary of Claimed Subject Matter" above.

B. Examiner's Rejection

The Examiner relies on AAPA as teaching all of the claimed elements except for "wherein the expanded key information is not stored in intermediate storage," "repeating the acts of obtaining, sorting, reorganizing and storing for at least a second plurality of data records," "merging the … reorganized data records" and "re-loading the merged plurality of reorganized data records into the database object." Office Action dated 15 May 2006 at pages 19-21. The Examiner relies on Matsuda to teach the recited "wherein the expanded key information is not stored in intermediate storage" act (Office Action dated 15 May 2006 at pages 21-22) and Ferguson for the remaining elements (Office Action dated 15 May 2006 at pages 22-23).

C. Analysis

All of Appellant's arguments with respect to AAPA and Matsuda presented in Section 5.C above are equally applicable and relevant to this rejection.

With respect to Ferguson, nowhere does this reference teach, describe or suggest data objects that have variable length keys. To the extent that Fergusson discusses compression of key information, it is key information (*i.e.*, substrings) created by Ferguson that are compressed and not the key information retrieved from the data objects being sorted. Ferguson at 12:47-57. In addition, key records created in accordance with Ferguson are used to generate substrings that are *stored back to disk – intermediate storage*. Independent claim 55 explicitly states that *expanded* key information is *not* returned to intermediate storage. Not only does the Specification identify this aspect of the invention as unique, it explicitly contrasts this operation with the prior art. Specification at pages 3, lines 9-17 (¶ 0005) and 6, lines 19-21 (¶ 0017). As a result, there is no teaching (explicit or implicit) in Ferguson to suggest or motivate the Examiner's alleged combination.

Again, the Examiner argues that because it would solve a known problem to combine the references, it is obvious to do so: "It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the claimed references because Ferguson's teaching would have allowed AAPA and Matsuda to require fewer storage system access and hence is generally faster … by merging by merging the reorganized data records and to sort key records first, and then build a tree based on keys extracted at intervals from sorted key records."[2] Office Action dated 15 May 2006 at pages 23 (last paragraph) to 24 (first paragraph).

As previously noted, such reasoning is not of the type or quality required to make a *prima facie* case of obviousness. (See discussion above.) For at least this reason, the Examiner has failed to state a *prima facie* case of obviousness under 35 U.S.C. 103(a), the elements of which are identified in Section 4 above.

[2] The fact that Ferguson describes a means to build a tree based on keys is irrelevant to the claimed invention. No such limitation is recited in independent claim 55 or any of its dependent claims, 56-63.

SUMMARY AND CONCLUSIONS

Because AAPA and Matsuda fail to teach or suggest (alone or in combination) all the claimed elements and, further, because there is absolutely no suggestion in either reference to motivate the Examiner's combination, independent claims 1, 16, 27 and 40 are not rendered obvious by the combination of AAPA and Matsuda. For at least these same reasons, dependent claims 2-12, 17-24, 28-37 and 41-52 are patentable over the combination of AAPA and Matsuda. Accordingly, Appellant respectfully requests that the Board withdraw the Examiner's section 103 rejection of claims 1-12, 16-24, 27-37 and 40-52.

There is, similarly, absolutely no basis to support the Examiner's combination of AAPA, Matsuda and Ferguson. Thus, independent claim 55 is not rendered obvious by the combination of AAPA, Matsuda and Ferguson. For at least these same reasons, dependent claims 13-15, 25, 26, 38-39 and 53, 54 and 56-63 are patentable over the combination of AAPA, Matsuda and Ferguson. Accordingly, Appellant respectfully requests that the Board withdraw the Examiner's section 103 rejection of claims 13-15, 25, 26, 38-39 and 53-63.

/Coe F. Miles, Ph.D., J.D./ Date: October 13, 2006
Reg. No. 38,559
Wong, Cabello, Lutsch, Rutherford & Brucculeri, L.L.P.
Customer No. 29855 Voice: 832-446-2418
20333 SH 249, Suite 600 Mobile: 713-502-5382
Houston, Texas 77070 Facsimile: 832-446-2458
Email: cmiles@counselIP.com

1. (Previously Presented) A data sort method, comprising:

obtaining a plurality of data records and, for each data record

extracting key information,

expanding the extracted key information, and

storing the expanded key information in a key record, wherein the expanded key information is not stored in intermediate storage;

sorting the plurality of key records based on the expanded key information;

reorganizing the plurality of data records to correspond to the order of the sorted plurality of key records; and

storing the reorganized plurality of data records without their associated expanded key information to a working storage.

2. (Original) The method of claim 1, wherein the act of obtaining comprises obtaining data records from one or more storage devices.

3. (Original) The method of claim 1, wherein the act of extracting comprises:

determining a starting location for a first key field; and

calculating the starting location of a subsequent key field based on the determined starting location of the first key field.

4. (Original) The method of claim 3, wherein the act of determining comprises obtaining the starting location of the first key field from a sort control card.

5. (Original) The method of claim 4, wherein the sort control card comprises a parameter list.

6. (Original) The method of claim 4, wherein the sort control card identifies a starting position for each key field in a record relative to a first key field of the record.

7. (Original) The method of claim 4, wherein the sort control card further indicates a data type for each key field in a record.

8. (Original) The method of claim 7, wherein the sort control card further indicates a sort order for each key field in a record.

9. (Original) The method of claim 1, wherein the act of expanding comprises adjusting each key field to a fixed length.

10. (Original) The method of claim 1, wherein the act of storing the expanded key information in a key record further comprises, associating a value with each key record that identifies the data record from which the expanded key information was extracted.

11. (Original) The method of claim 10, wherein the act of storing the expanded key information in a key record does not comprise storing a data field from the data record associated with the key record.

12. (Original) The method of claim 1, wherein the working storage comprises one or more direct access storage devices.

13. (Original) The method of claim 1, further comprising repeating the acts of obtaining, sorting, reorganizing and storing for at least a second plurality of data records.

14. (Original) The method of claim 13, further comprising merging the two or more plurality of reorganized data records.

15. (Original) The method of claim 14, wherein the act of obtaining a plurality of data records comprises obtaining a plurality of DB2 data records and the act of merging further comprises reloading the merged plurality of reorganized data records into the DB2 data object.

16. (Previously Presented) A method for sorting an object, the object having a plurality of records, each record having a plurality of key fields at least one of which is a variable length key field, the method comprising:

retrieving a plurality of records of an object;

extracting each key field in a record into a fixed length component of a corresponding key record, wherein the key record is not stored in intermediate storage;

sorting the plurality of key records based on the extracted fixed length components;

reordering the plurality of records based on the sorted plurality of key records; and

storing the reordered plurality of records in an intermediate storage, wherein the act of storing does not include storing fixed length components of a key record.

17. (Original) The method of claim 16, wherein the act of retrieving comprises retrieving data records from one or more storage devices.

18. (Original) The method of claim 16, wherein the act of extracting comprises using a sort control card to determine the starting locations of each of the plurality of key fields.

19. (Original) The method of claim 18, wherein the sort control card comprises a parameter list.

20. (Original) The method of claim 18, wherein the sort control card further indicates a data type for each key field in a record.

21. (Original) The method of claim 20, wherein the sort control card further indicates a sort order for each key field in a record.

22. (Original) The method of claim 16, wherein the act of extracting each key field in a record into a fixed length component comprises expanding each key field to a maximum length.

23. (Original) The method of claim 16, wherein the act of extracting further comprises associating a value with each key record that identifies the data record from which the key was extracted.

24. (Original) The method of claim 23, wherein the act of extracting each key field in a record into a fixed length component of a corresponding key record does not comprise storing data fields associated with the data record in the key record.

25. (Original) The method of claim 16, further comprising repeating the acts of retrieving, extracting, sorting, reordering and storing for at least a second plurality of records.

26. (Original) The method of claim 25, further comprising merging the two or more plurality of reordered records.

27. (Previously Presented) A program storage device, readable by a programmable control device, comprising instructions stored on the program storage device for causing the programmable control device to:

obtain a plurality of data records from a data object and, for each data record (1) extract key information, (2) expand the extracted key information, and (3) store the expanded key information in a key record, wherein the expanded key information is not stored in intermediate storage;

sort the plurality of key records based on the expanded key information;

reorganize the plurality of data records to correspond to the order of the sorted plurality of key records; and

store the reorganized plurality of data records without their associated expanded key information to a working storage.

28. (Original) The program storage device of claim 27, wherein the instructions to obtain comprise instructions to obtain data records from one or more storage devices.

29. (Original) The program storage device of claim 27, wherein the instructions to extract comprise instructions to:

determine a starting location for a first key field; and

calculate the starting location of a subsequent key field based on the determined starting location of the first key field.

30. (Original) The program storage device of claim 29, wherein the instructions to determine comprise instructions to obtain the starting location of the first key field from a sort control card.

31. (Original) The program storage device of claim 30, wherein the sort control card comprises a parameter list.

32. (Original) The program storage device of claim 30, wherein the sort control card identifies a starting position for each key field in a record relative to a first key field of the record.

33. (Original) The program storage device of claim 30, wherein the sort control card further indicates a data type for each key field in a record.

34. (Original) The program storage device of claim 33, wherein the sort control card further indicates a sort order for each key field in a record.

35. (Original) The program storage device of claim 27, wherein the instructions to expand comprise instructions to adjust each key field to a fixed length.

36. (Original) The program storage device of claim 27, wherein the instructions to store the expanded key information in a key record further comprise instructions to associate a value with each key record that identifies the data record from which the expanded key information was extracted.

37. (Original) The program storage device of claim 36, wherein the instructions to store the expanded key information in a key record do not comprise instructions to store a data field of the data record associated with the key record.

38. (Original) The program storage device of claim 27, wherein the instructions to obtain, sort, reorganize and store are performed for at least a second plurality of data records.

39. (Original) The program storage device of claim 38, further comprising instructions to merge the two or more plurality of reorganized data records.

40. (Previously Presented) A sorting system comprising:

memory means for storing a data object and instructions; and

processing means, communicatively coupled to the memory means, for executing the instructions to cause the processing means to –

obtain a plurality of data records from the data object and, for each data record (1) extract key information, (2) expand the extracted key information, and (3) store the expanded key information in a key record, wherein the expanded key information is not stored in intermediate storage

sort the plurality of key records based on the expanded key information,

reorganize the plurality of data records to correspond to the order of the sorted plurality of key records, and

store the reorganized plurality of data records without their associated expanded key information to a working storage.

41. (Original) The sorting system of claim 40, wherein the memory means comprises two or more storage devices coupled by a communications network.

42. (Original) The sorting system of claim 41, wherein the data object comprises a first plurality of records stored on a first storage device and a second plurality of records stored on a second storage device.

43. (Original) The sorting system of claim 40, wherein the processing means comprises two or more communicatively coupled computer processors.

44. (Original) The sorting system of claim 40, wherein the instructions to extract comprise instructions to:

determine a starting location for a first key field; and

calculate the starting location of a subsequent key field based on the determined starting location of the first key field.

45. (Original) The sorting system of claim 44, wherein the instructions to determine comprise instructions to obtain the starting location of the first key field from a sort control card.

46. (Original) The sorting system of claim 45, wherein the sort control card comprises a parameter list.

47. (Original) The sorting system of claim 45, wherein the sort control card identifies a starting position for each key field in a record relative to a first key field of the record.

48. (Original) The sorting system of claim 45, wherein the sort control card further indicates a data type for each key field in a record.

49. (Original) The sorting system of claim 48, wherein the sort control card further indicates a sort order for each key field in a record.

50. (Original) The sorting system of claim 40, wherein the instructions to expand comprise instructions to adjust each key field to a fixed length.

51. (Original) The sorting system of claim 40, wherein the instructions to store the expanded key information in a key record further comprise instructions to associate a value with each key record that identifies the data record from which the expanded key information was extracted.

52. (Original) The sorting system of claim 51, wherein the instructions to store the expanded key information in a key record does not comprise instructions to store data fields associated with the data record in the key record.

53. (Original) The sorting system of claim 40, wherein the instructions to obtain, sort, reorganize and store are performed for at least a second plurality of data records.

54. (Original) The sorting system of claim 53, further comprising instructions to merge the two or more plurality of reorganized data records.

55. (Previously Presented) A data sort method, comprising:

obtaining a plurality of data records from a database object, for each of the plurality of data records –

extracting key information,

expanding the extracted key information, and

storing the expanded key information in a key record, wherein the expanded key information is not stored in intermediate storage;

sorting the plurality of key records based on the expanded key information;

reorganizing the plurality of data records to correspond to the order of the sorted plurality of key records;

storing the reorganized plurality of data records without their associated expanded key information in a working storage;

repeating the acts of obtaining, sorting, reorganizing and storing for at least a second plurality of data records;

merging the at least two plurality of reorganized data records; and

re-loading the merged plurality of reorganized data records into the database object.

56. (Original) The data sort method of claim 55, wherein the act of extracting comprises obtaining the starting location of a first key field in a data record from a sort control card.

57. (Original) The data sort method of claim 56, wherein the sort control card identifies a starting position for each key field in a record relative to a first key field of the record.

58. (Original) The data sort method of claim 56, wherein the sort control card further indicates a data type for each key field in a record.

59. (Original) The data sort method of claim 58, wherein the sort control card further indicates a sort order for each key field in a record.

60. (Original) The data sort method of claim 58, wherein the sort control card comprises a parameter list.

61. (Original) The data sort method of claim 55, wherein the act of expanding comprises adjusting each key field to a fixed length.

62. (Original) The data sort method of claim 55, wherein the act of storing the expanded key information in a key record further comprises, associating a value with each key record that identifies the data record from which the expanded key information was extracted.

63. (Original) The data sort method of claim 62, wherein the act of storing the expanded key information in a key record does not comprise storing a data field from the data record associated with the key record.

64. (Withdrawn) A memory for storing a data structure for access by a sort program being executed on a data processing system, the data structure comprising:

a starting location indicator for a first key field in a data object record;

a maximum length indicator associated with the first key field;

a field type indicator associated with the first key field;

a sort-order indicator associated with the first key field; and

starting, maximum length, field type and sort-order indicators for at least a second key field.

65. (Withdrawn) The memory of claim 64, wherein the data structure further comprises starting location, length, field type and sort-order indicators for at least one data field in the data object record, wherein the sort-order indicator is set to a value indicative of a non-key field.

66. (Withdrawn) The memory of claim 64, wherein the starting location indicator for each field subsequent to a variable length field is set to a value to indicate that a prior field is a variable length field.

67. (Withdrawn) The memory of claim 64, wherein the field type indicator indicates a field type selected from the group consisting of a floating point number, an integer, a Boolean value, a binary value and a character value.

68. (Withdrawn) The memory of claim 67, wherein the field type indicator further comprises an indication of whether a filed is a variable length field or fixed length field.

69. (Withdrawn) The memory of claim 64, wherein the sort-order indicator indicates an ascending sort or a descending sort order.

Evidence Appendix

< NONE >

Related Proceedings Appendix

< NONE >